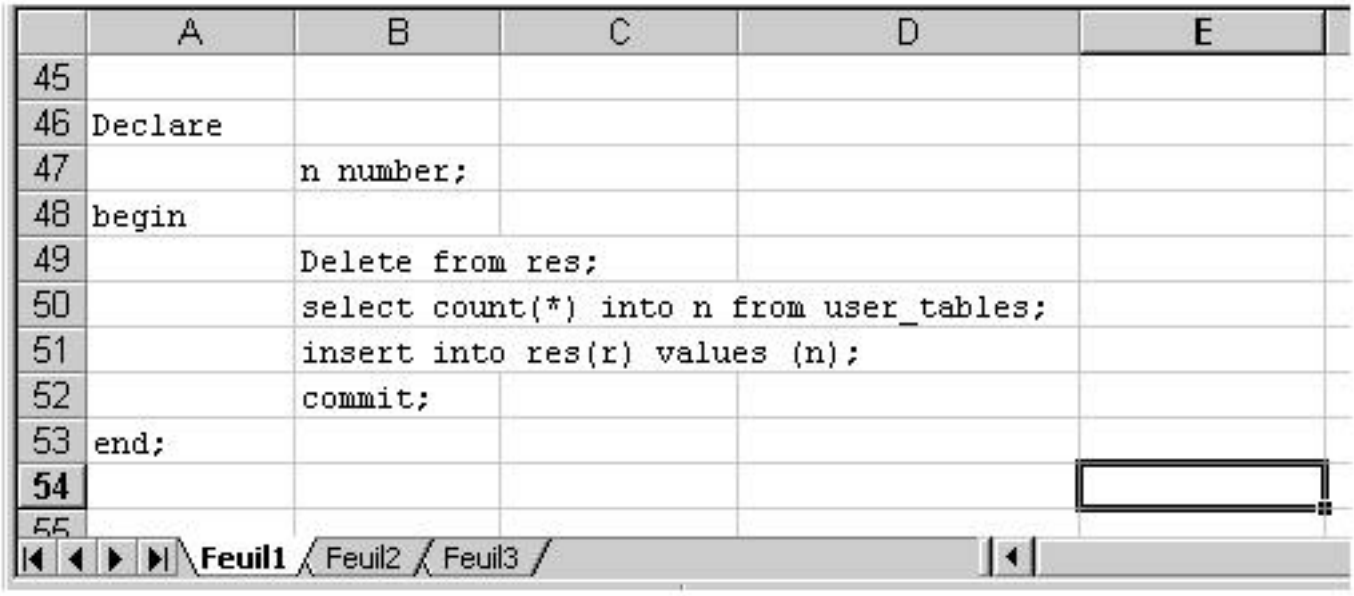
Mise en œuvre du langage PL.SQL



	A	B	C	D	E
45					
46	Declare				
47		n number;			
48	begin				
49		Delete from res;			
50		select count(*) into n from user_tables;			
51		insert into res(r) values (n);			
52		commit;			
53	end;				
54					

Feuil1 Feuil2 Feuil3

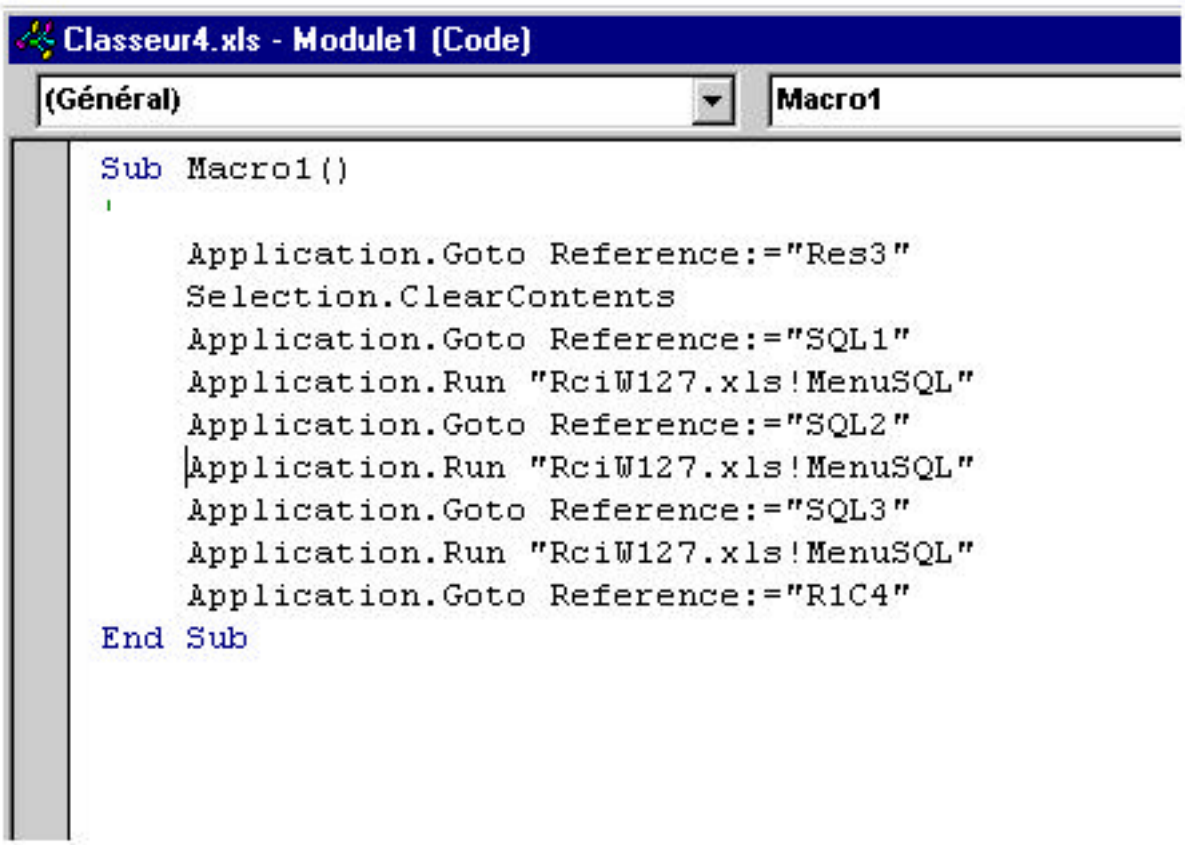
Il est possible avec RCITools Oracle/Excel de soumettre des ordres PL.SQL.

L'utilisation conjointe de cette possibilité avec celle de lire et d'écrire très facilement des fichiers ascii comportant du SQL, a été conçue dans le but d'offrir un environnement facilitant au maximum la manipulation de blocs PL.SQL.

Automatisation de Rci.Tools

Il est possible de déclencher automatiquement le lancement de requêtes à RCI.Tools, exactement de la même façon que si l'on utilisait le sous-menu DoSQL.

Voici un exemple de sous-programme VBA (Visual Basic Applications) enchaînant une suite de requêtes SQL localisées dans des groupes de cellules nommés SQL1, SQL2, SQL3 et SQL4 :



Notez la façon d'appeler depuis un module macro de votre feuille de calcul, la procédure MenuSQL de la feuille RciW127.xls.

Mise en oeuvre de RciTools Oracle/Excel et du langage VBA

- VBA (Visual Basic Applications) est le langage commun aux outils bureautique de Microsoft, dont Microsoft Excel.

- Le logiciel RciTools Oracle/Excel comporte un certain nombre de "primitives", permettant d'effectuer de manière très simple des requêtes à Oracle, depuis Excel VBA.

Primitives disponibles :

RciDoLogin ***(byVal xUser as String, byVal xPassword as String, byVal xServer as String)***
résultat Integer : 0 si succès de la connexion, sinon le code erreur Oracle

Action, établissement d'une connexion vers un serveur Oracle
La connexion reste ouverte jusqu'au changement de connexion ou au fait de quitter Excel.

RCIdoSelectSQL ***(byVal xSQL as String,***
byVal xRow as Integer, byVal xColumn as Integer, byVal xMaxRows as Integer)
résultat Integer : 0 si succès de la requête, sinon le code erreur Oracle

Action: Lancement de la requête SQL fournie dans l'argument xSQL,
Le résultat de la requête est placée dans la feuille active, à la position (xRow, xColumn). Si le nombre de lignes renvoyées par la requete est supérieur à la valeur xMaxRows, un dialogue est affiché à l'utilisateur avec mention du nombre effectif de ligne.

RCIgetSQL ***(byVal xFile as String)***
résultat String :
"error" si une erreur s'est produite
- le contenu du fichier si celui ci a pu être trouvé et lu

Action: Lecture du fichier passé en argument et renvoi du contenu de celui-ci en tant que chaîne de caractères.

RCIstrReplaceOne ***(byVal xString as String, byVal xNewValue as String)***
résultat String :
"error" si une erreur s'est produite
- le contenu de la chaine remplacée

Action: Recherche dans la chaîne xString l'occurence ou les différentes occurences de la sous-chaîne "%1", et remplacement par la sous-chaîne xNewvalue.

Exemple 1 - Connexion à Oracle

- Objectif

Tester une connection à une base Oracle et afficher un message en fonction du résultat de la connexion.

- Source du programme

```
'Copyright RCI Informatique SA 1998
'Sample VBAsmp01.xls

Public RCIdoLogin As String

Sub test1()
    '-- Declarations
    Dim wUser, wPwd, wServer As String
    Dim r As Integer

    '-- Routine
    Call RciInit 'inits

    wServer = InputBox(Prompt:="Server ?")
    r = Run(RCIdoLogin, "Scott", "Tiger", wServer)
    If r <> 0 Then
        MsgBox ("Error - " & r)
        Exit Sub
    Else
        MsgBox ("Connected to Scott at " & wServer)
    End If
 End Sub

Sub RciInit()
    ModuleName = "rciw127.xls"
    RCIdoLogin = ModuleName & "!RCIdoLogin"
End Sub
```

Exemple 2 - Requête simple, renvoyant une valeur

- Objectif

Exécuter une requête SQL stockée dans une variable de type String

- Source du programme

```
'Copyright RCI Informatique SA 1998
'Sample VBAsmp02.xls

Public RCIdoSelectSQL As String

Sub test1()
    '-- Declarations
    Dim r As Integer
    Dim wSQL As String

    '-- Routine
    Call RciInit 'inits

    wSQL = "Select count(*) from user_tables"
    r = Run(RCIdoSelectSQL, wSQL, 1, 1, 1) '-- To Cell A1, one value
    If r <> 0 Then MsgBox ("Error - " & r): Exit Sub

    MsgBox "Result = " & ActiveSheet.Cells(2, 1)
End Sub

Sub RciInit()
    ModuleName = "rciw127.xls"
    RCIdoSelectSQL = ModuleName & "!RCIdoSelectSQL"
End Sub
```

Exemple 3 - Listage des tables du User courant, et pour chaque table détermination du nombre de lignes

- Objectif

Exécuter une requête SQL stockée dans une variable de type String

- Source du programme

```
'Copyright RCI Informatique SA 1995/1998
'Sample VBAsmp03.xls

Public RCIdoSelectSQL As String
Public RCIgetSQL As String
Public RCIstrReplaceOne As String

Sub test1()
    Call RciInit 'initialisations routines
    'login depuis le sous-programme, sans affichage de dialogue
    ActiveWindow.ScrollRow = 1: Columns("A:B").ClearContents
    'chargement de la requête SQL de liste des tables
    wSQL = Run(RCIgetSQL, "c:\RciTools\samples\VBA002T1.sql")
    If wSQL = "error" Then Exit Sub
    'exécution de la requête, résultat en ligne 1, colonne 1, max 10 lignes
    r = Run(RCIdoSelectSQL, wSQL, 1, 1, 10): If r <> 0 Then Exit Sub
    k = 1
    Do
        k = k + 1 'on ne traite pas le titre
        wTable = ActiveSheet.Cells(k, 1): If wTable = "" Then Exit Do
        'Lecture de la requête SQL depuis le disque
        wSQL = Run(RCIgetSQL, "c:\RciTools\samples\VBA002T2.sql") 'chgt requête
            If wSQL = "error" Then Exit Sub
        'remplacement de la variable %1 par le nom de la table
        wSQL = Run(RCIstrReplaceOne, wSQL, wTable)
        If wSQL = "error" Then Exit Sub
        r = Run(RCIdoSelectSQL, wSQL, 1, 3, 1): If r <> 0 Then Exit Sub
        ActiveSheet.Cells(k, 2) = ActiveSheet.Cells(2, 3)
    Loop
    ActiveSheet.Cells(1, 3) = "": ActiveSheet.Cells(2, 3) = "" 'effacement zone
temp
End Sub

Sub RciInit()
    ModuleName = "rciw127.xls"
    RCIdoSelectSQL = ModuleName & "!RCIdoSelectSQL"
    RCIgetSQL = ModuleName & "!RCIgetSQL"
    RCIstrReplaceOne = ModuleName & "!RCIstrReplaceOne"
End Sub
```